

---

**Flash memory and control method****Patent Assignee: SAMSUNG ELECTRONICS CO LTD****Inventors: KIM B S**

Patent Family							
Patent Number	Kind	Date	Application Number	Kind	Date	Week	Type
CN 1290014	A	20010404	CN 2000126378	A	20000912	200221	B
KR 2001029171	A	20010406	KR 9941835	A	19990929	200379	
US 6587915	B1	20030701	US 2000671176	A	20000928	200345	

**Priority Applications (Number Kind Date):** KR 9941835 A ( 19990929)

Patent Details					
Patent	Kind	Language	Page	Main IPC	Filing Notes
CN 1290014	A			G11C-016/06	
KR 2001029171	A		1	G11C-016/00	
US 6587915	B1			G11C-016/00	

**Abstract:**

KR 2001029171 A

NOVELTY A flash memory is provided to write or read a data by a block unit as efficiently managing a spirit between a logical address and a physical address of a block.

DETAILED DESCRIPTION The flash memory includes a plurality of data blocks for writing a data, a plurality of spare blocks, a map block and a unit. The plurality of spare blocks are in empty for writing a renewal data of the data block. The map block designates a real position to which a data is written. The unit is consisted of a head block recording an information for the flash memory and an information for the unit. The unit is used as a basic unit in an erasing operation. Information for a logical unit number(LUN), a degree of abrasion and the size of a block are written to the head block.

Derwent World Patents Index

© 2004 Derwent Information Ltd. All rights reserved.

Dialog® File Number 351 Accession Number 14334839

# (19) 대한민국특허청(KR) (12) 공개특허공보(A)

(51) Int. Cl. <sup>8</sup> G11C 16/00	(11) 공개번호 (43) 공개일자	특2001-0029171 2001년04월06일
(21) 출원번호	10-1999-0041835	
(22) 출원일자	1999년09월29일	
(71) 출원인	삼성전자 주식회사	윤종용
(72) 발명자	경기 수원시 팔달구 매탄3동 416 김범수	
(74) 대리인	서울특별시강남구신사동599-4압구정빌딩6층 윤의섭	

심사청구 : 없음

## (54) 플래시 메모리와 그 제어 방법

### 요약

본 발명은 플래시 메모리와 그 제어 방법에 관한 것으로,

본 발명의 플래시 메모리가, 데이터를 기록하기 위한 복수개의 데이터 블록 : 상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록 ; 데이터가 기록된 실제 위치를 지정하는 맵 블록 : 플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성되어,

데이터 블록의 데이터를 갱신하고자 할 경우 먼저 갱신할 데이터를 동일 유니트내의 비어있는 예비 블록에 기록한 후 맵 블록의 사상정보를 갱신하고 이전 블록의 상태정보를 갱신함으로써, 사용자가 동일한 주소를 이용하여 데이터에 접근할 수 있어, 블록을 갱신할 때마다 유니트를 지우지 않아도 되기 때문에, 삭제 횟수를 감소시켜서 데이터 기록 및 갱신의 효율을 높인다는 데 그 효과가 있다.

### 대표도

### 도3

### 색인어

플래시메모리, 제어방법, 사상, 삭제 유니트, 블록

### 명세서

### 도면의 간단한 설명

도 1 의 (a)는 일반적인 고정 사상 방법을 설명하기 위한 도면이고, (b)는 일반적인 임의 사상 방법을 설명하기 위한 도면이고, (c)는 본 발명에 따른 혼합 사상 방법을 설명하기 위한 도면,

도 2 는 일반적인 플래시 메모리와 그 주변 회로를 도시한 블록도,

도 3 은 본 발명에 따른 플래시 메모리와 그 제어 방법을 설명하기 위한 도면으로, (a)는 LBN-to-LUN 맵 테이블의 구성도, (b)는 LUN-to-PUN 맵 테이블의 구성도, (c)(d)는 플래시 메모리내 유니트의 구성도,

도 4 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 사상 동작의 흐름도로서,

도 5 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 블록 할당 동작의 흐름도,

도 6 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 독출 연산의 흐름도,

도 7 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 삭제 연산의 흐름도,

도 8 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 갱신 연산의 흐름도,

도 9 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 재생 연산의 흐름도,

도 10 은 도 8 의 갱신 연산 수행중 에러 발생시 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 회복 동작의 상태도,

도 11 은 도 9 의 재생 연산 수행중 에러 발생시 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 회복 동작의 상태도,

도 12 는 도 8 의 갱신 연산을 설명하기 위한 도면으로, (a)는 LBN-to-LUN 맵 테이블의 구성도, (b)는 LUN-to-PUN 맵 테이블의 구성도, (c)(d)는 플래시 메모리내 유니트의 구성도,

도 13 은 도 9 의 재생 연산을 설명하기 위한 도면으로, (a)는 LBN-to-LUN 맵 테이블의 구성도, (b)는 LUN-to-PUN 맵 테이블의 구성도, (c)(d)(e)는 플래시 메모리내 유니트의 구성도이다.

\* 도면의 주요 부분에 대한 부호의 명칭 \*

- |            |             |
|------------|-------------|
| 1 : 중앙처리장치 | 2 : 플래시 메모리 |
| 3 : 램      | 4 : 롬       |

## 발명의 상세한 설명

### 발명의 목적

#### 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 플래시 메모리와 그 제어 방법에 관한 것이다.

보다 상세하게는, 블록의 논리적 주소와 물리적 주소간의 사상을 효율적으로 관리하여 블록 단위로 데이터를 기록하거나 독출하도록 되어진 플래시 메모리와 그 제어 방법에 관한 것이다.

1980년대 초 인텔에 의해서 처음으로 개발된 플래시 메모리는 EPROM의 일종이지만, 일반적인 EPROM과는 달리 프로그램을 위한 하드웨어가 별도로 필요없으며 소프트웨어로 데이터를 삭제하거나 기록할 수 있다는 특성을 가지고 있다.

이같이 플래시 메모리는 비휘발성이면서 프로그램 가능하다는 특성 때문에 최근에 이동전화나 PDA 같은 이동기기를 비롯한 내장형 시스템에서 널리 사용되고 있다.

플래시 메모리는 기존의 RAM 이나 다른 휘발성 저장장치, 마그네틱 디스크 등과 마찬가지로 특정 위치에 저장된 데이터를 임의로 접근할 수 있지만, 데이터를 수정하거나 삭제하는 방법이 다르다.

즉, 초기화된 플래시 메모리의 소정 블록에 일단 데이터를 한 번 기록하고 나면 나중에 이를 삭제하고자 할 경우 그 블록이 포함된 유니트를 모두 삭제해야 한다.

여기서 물리적으로 연속된 주소를 갖는 바이트들을 블록이라 하는데, 상기 블록은 플래시 메모리에 대한 연산의 기본 단위로 사용된다. 또한 상기 복수개의 블록들로 구성된 유니트는 플래시 메모리에서 물리적으로 한번에 삭제될 수 있는 삭제 연산의 기본 단위이다.

따라서 플래시 메모리는 RAM과 같이 바이트 단위로 읽고 쓰기 보다는 마치 디스크처럼 블록 단위로 접근하는 경우가 많다.

플래시 메모리를 블록 디바이스처럼 사용할 수 있게 해주는 소프트웨어를 일반적으로 플래시 변환 계층(FTL : Flash Translation Layer)이라고 한다.

대표적인 FTL 제품으로는 M-System사의 TrueFFS[FTL97]와 Datalight사의 FlashFX[FX99] 등이 있다.

이들 소프트웨어의 여러 구성 요소중에서 특히 블록 재사상을 지원하는 부분을, TrueFFS[FX99]에서는 FTL, FlashFX에서는 VBX라 하며, M-System사의 FTL은 PC Card 표준의 저장 포맷으로 채택되기도 했다.

플래시메모리를 블록 단위로 접근하려면 일단 플래시 메모리를 유니트보다 작은 일정한 크기의 블록으로 분할한다.

이렇게 분할된 블록에 순차적으로 부여된 번호를 물리적 블록 번호(PBN : Physical Block Number)라 하며, 사용자가 생각하는 블록의 가상 번호를 논리적 블록 번호(LBN : Logical Setor Number)라 한다.

상기 논리적 블록 번호(LBN)와 물리적 블록 번호(PBN) 사이의 사상을 제공하는 방식은 크게 고정 사상과 임의 사상으로 구분할 수 있다.

#### a. 고정 사상

고정 사상은 논리적 블록 번호(LBN)와 물리적 블록 번호(PBN) 사이의 관계가 일정하게 정해져 있는 방식이다.

별도의 헤더를 사용하지 않는 매우 단순한 경우라면  $LBN=PBN$  이라 할 수 있다.

도 1(a) 는 일반적인 고정 사상을 설명하기 위한 도면이고, 도 2 는 일반적인 플래시 메모리와 그 주변 회로를 도시한 블록도로서, 도 1(a) 및 도 2 에 도시된 바와 같이 플래시 메모리(2)내 유니트의 블록 2 에 기록된 데이터를 갱신하려면 블록 1,3,4의 데이터를 일단 독출하여 RAM(3)에 기록하고 이들 블록이 속한 유니트의 데이터를 모두 삭제한 뒤, 상기 RAM에 기록된 블록 1,3,4의 이전 데이터를 독출하여 다시 블록 1,3,4에 기록하고, 블록 2에는 새로운 데이터를 기록한다.

따라서, 블록의 데이터를 삭제하고 재기록하는 데 많은 시간이 걸리게 될 뿐만 아니라, 블록의 데이터를 삭제한 직후에 고장이 발생하면 블록 1,2,3,4의 데이터를 모두 잃을 수 있다.

즉, 고정 사상은 복잡한 사상 정보를 유지하기 위한 저장 공간과 알고리즘이 필요하지 않고 코드 크기가 작다는 장점이 있으나, 삭제 연산이나 갱신 연산의 성능이 좋지 않다는 단점이 있다.

#### b. 임의 사상

임의 사상은 기존의 FTL 제품에서 주로 사용되는 것으로, 논리적 블록 번호(LBN)와 물리적 블록 번호(PBN) 사이의 관계가 일정하게 정해져 있지 않다.

즉, 비어 있는 물리적 블록에 임의의 논리적 블록이 할당된다. MAP(LBN)=PBN.

논리적 유니트 번호(LUN)와 물리적 유니트 번호(PUN)간의 사상 정보, 논리적 블록 번호(LBN)와 물리적 블록 번호(PBN)간의 사상 정보는 RAM(3)와 플래시 메모리(2)에 저장된다.

RAM(3)상의 맵 테이블에는 물리적 유니트 번호(PUN)와 논리적 유니트 번호(LUN)간의 사상 정보가 기록되어 있어, 논리적 주소를 물리적 주소로 변환하는데 사용되는데 사용되는 바, 삭제 연산이나 갱신 연산 혹은 재생 연산으로 인해 사상 관계가 변하면 상기 맵 테이블도 갱신되어 동일한 논리적 주소로 계속 접근할 수 있다.

도 1(b)는 일반적인 임의 사상을 설명하기 위한 도면으로, 도 1(b)에 도시된 바와 같이 중앙처리장치(1)가 독출 연산을 수행하는 경우, 사용자가 요청한 논리적 블록 번호(LBN)는 맵 테이블을 통해 물리적 블록 번호(PBN)로 변환된다.

기록 연산도 상기 독출 연산과 마찬가지로 해당 블록에 접근하기 위해 먼저 논리적 블록 번호(LBN)를 물리적 블록 번호(PBN)로 변환하는데, 이때 해당 블록에 이미 데이터가 기록되어 있는 상태라면 플래시 메모리의 특성상 그 블록이 속한 유니트의 데이터를 모두 삭제한 후, 갱신하지 않아야 할 블록의 데이터를 갱신해야 할 블록의 데이터를 삭제된 유니트에 다시 재기록하거나,

또는 새로운 유니트에 재기록하고 이전 유니트는 삭제된 것으로 표시한다.

이때 상기 맵 테이블은 상기 논리적 블록 번호(LBN)에 대한 물리적 블록 번호(PBN)가 갱신된다.

또한, 유니트에 사용되지 않는 블록들이 많아지면 메모리 사용율이 낮아지므로, 사용되는 부분만을 다른 유니트로 이동시키는데, 유니트에 논리적 번호를 매기고 이것을 물리적 블록으로 변환함으로써, 유니트의 데이터를 이동하더라도 동일한 논리적 유니트 번호(LUN)로 접근할 수 있다.

상기에서 살펴본 바와 같이 종래 플래시 메모리에 이용되는 임의 사상 방법은 맵 테이블을 이용하여 플래시 메모리상에 데이터를 기록하거나, 플래시 메모리에 기록된 데이터를 독출하는 방법을 제시하고 있다.

종래 플래시 메모리에 이용되는 임의 사상 방법은 논리적 주소를 임의의 물리적 주소로 변환하는데 필요한 사상 정보를 유지하기 위해 별도의 저장 공간이 필요하며, 데이터를 기록하거나 갱신해야 하는데 많은 시간이 소요될 뿐만 아니라 복잡한 사상 구조와 알고리즘을 오류없이 구현하거나 이식하는 것이 어렵다는 문제점이 있었다.

#### **발명이 이루고자하는 기술적 과제**

이에 본 발명은 상기와 같은 문제점을 해결하기 위해 안출된 것으로, 블록의 논리적 주소와 물리적 주소간의 사상을 효율적으로 관리하여 블록 단위로 데이터를 기록하거나 독출하도록 되어진 플래시 메모리와 그 제어 방법을 제공하는 데 그 목적이 있다.

상기와 같은 목적을 달성하기 위한 본 발명에 따른 플래시 메모리가,

데이터를 기록하기 위한 복수개의 데이터 블록 ;

상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록 ;

데이터가 기록된 실제 위치를 지정하는 맵 블록 ;

플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성된 것을 특징한다.

또한 상기와 같은 목적을 달성하기 위한 본 발명에 따른 플래시 메모리의 제어방법은,

데이터를 기록하기 위한 복수개의 데이터 블록 ; 상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록 ; 데이터가 기록된 실제 위치를 지정하는 맵 블록 ; 플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성된 플래시 메모리에 있어서,

논리적 블록이 속하는 논리적 유니트를 탐색하는 과정 ;

상기 논리적 유니트에 대한 물리적 유니트를 탐색하는 과정 ;

상기 물리적 유니트의 맵 블록에서 상기 논리적 블록에 대한 물리적 블록의 위치를 탐색하는 과정으로 사상 동작을 제어하는 것을 특징으로 한다.

#### **발명의 구성 및 작용**

이에 상기와 같이 구성된 본 발명의 플래시 메모리와 그 제어방법에 따른 일실시예를 살펴보도록 한다.

도 3 (c)(d)는 본 발명에 따른 플래시 메모리내 유니트의 구성도이다.

도 3(c)(d)에 도시된 바와 같이 본 발명에 따른 플래시 메모리는, 데이터를 기록하기 위한 복수개의 데이터 블록 ; 상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록 ; 데이터가 기록된 실제 위치를 지정하는 맵 블록 ; 플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성된다.

여기서 상기 유니트는 삭제 연산의 기본 단위로 사용되며, 상기 헤더 블록에는 논리적 유니트 번호(LUN), 마모도, 블록의 크기에 대한 정보를 기록하고, 상기 맵 블록에는 논리적 블록 번호(LBN), 블

력의 상태 정보, 갱신된 블록의 위치 정보를 기록한다.

도 4 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 사상 동작의 흐름도이다.

도 4 에 도시된 바와 같이 본 발명의 방법은, 데이터를 기록하기 위한 복수개의 데이터 블록; 상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록; 데이터가 기록된 실제 위치를 지정하는 맵 블록; 플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성된 플래시 메모리에 있어서, 논리적 블록이 속하는 논리적 유니트를 탐색하는 과정(S11); 상기 논리적 유니트에 대한 물리적 유니트를 탐색하는 과정(S12); 상기 물리적 유니트의 맵 블록에서 상기 논리적 블록에 대한 물리적 블록의 위치를 탐색하는 과정(S13)으로 사상 동작을 제어한다.

도 5 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 블록 할당 동작의 흐름도이다.

도 5 에 도시된 바와 같이 본 발명의 방법은, 상기 유니트의 맵 블록을 검사하여 상기 유니트에 예비 블록이 있는가를 판단하는 제 21 단계(S21); 상기 제 21 단계(S21)에서 판단한 결과 상기 유니트에 예비 블록이 없으면 상기 유니트에 대해 재생연산을 수행하는 제 22 단계(S22); 상기 재생된 유니트의 맵 블록을 검사하여 상기 재생된 유니트에 예비 블록이 있는가를 판단하는 제 23 단계(S23); 상기 제 21 단계(S21)에서 판단한 결과 상기 유니트에 예비 블록이 있거나 또는 상기 제 23 단계(S23)에서 판단한 결과 상기 재생된 유니트에 예비 블록이 있으면, 상기 맵 블록에 데이터 블록의 논리적 블록 번호(LBN)와 상태를 기록하는 제 24 단계(S24); 상기 제 23 단계(S23)에서 판단한 결과 상기 재생된 유니트에 예비 블록이 없으면 에러 처리하는 제 25 단계(S25)로 할당 연산을 제어한다.

도 6 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 독출 연산의 흐름도이다.

도 6 에 도시된 바와 같이 본 발명의 방법은, 해당 데이터 블록이 속한 유니트를 탐색하는 제 31 단계(S31); 상기 유니트내에 해당 데이터 블록이 존재하는가를 판단하는 제 32 단계(S32); 상기 제 32 단계(S32)에서 판단한 결과 해당 데이터 블록이 존재하면, 상기 해당 데이터 블록에서 데이터를 독출하는 제 33 단계(S32); 상기 제 32 단계(S32)에서 판단한 결과 해당 데이터 블록이 존재하지 않으면 에러 처리하는 제 34 단계(S34)로 독출 연산을 제어한다.

도 7 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 삭제 연산의 흐름도이다.

도 7 에 도시된 바와 같이 본 발명의 방법은, 해당 데이터 블록이 속한 유니트를 탐색하는 제 41 단계; 상기 유니트의 맵 블록에 기록된 해당 데이터 블록의 상태를 "삭제중"으로 갱신하는 제 42 단계(S42)로 삭제 연산을 제어한다.

도 8 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 갱신 연산의 흐름도이다.

도 8 에 도시된 바와 같이 본 발명의 방법은, 해당 데이터 블록이 속한 유니트를 결정하는 제 51 단계(S51); 상기 유니트내에 해당 데이터 블록이 존재하는가를 판단하는 제 52 단계(S52); 상기 제 52 단계(S52)에서 판단한 결과 해당 데이터 블록이 존재하면, 맵 블록을 검사하여 해당 데이터 블록의 상태가 "할당됨"인가를 판단하는 제 53 단계(S53); 상기 제 53 단계(S53)에서 판단한 결과 해당 블록이 "할당됨" 상태이면 해당 데이터 블록 상태를 "기록중"으로 갱신하는 제 54 단계(S54); 해당 데이터 블록에 데이터를 기록하는 제 55 단계(S55); 해당 데이터 블록 상태를 "기록됨"으로 갱신하는 제 56 단계(S56); 상기 제 53 단계(S53)에서 판단한 결과 해당 데이터 블록이 "할당됨" 상태이면 블록의 상태를 "삭제중"으로 갱신하는 제 57 단계(S57); 상기 유니트내에서 예비 블록을 탐색하여 할당하는 제 58 단계(S58); 새롭게 할당된 블록의 상태를 "기록중"으로 갱신하는 제 59 단계(S59); 상기 새롭게 할당된 블록의 위치 정보를 상기 맵 블록에 기록하는 제 60 단계(S60); 상기 새롭게 할당된 블록에 데이터를 기록하는 제 61 단계(S61); 상기 새롭게 할당된 블록의 상태를 "기록됨"으로 갱신하는 제 62 단계(S62); 상기 해당 데이터 블록의 상태를 "삭제됨"으로 갱신하는 제 63 단계(S63); 상기 제 52 단계(S52)에서 판단한 결과 해당 데이터 블록이 존재하지 않으면 에러 처리하는 제 64 단계(S64)로 갱신 연산을 제어한다.

도 9 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 재생 연산의 흐름도이다.

도 9 에 도시된 바와 같이 본 발명의 방법은, 재생할 유니트(EU)와 이동 유니트(TU)를 결정하는 제 71 단계(S71); 상기 이동 유니트(TU)의 삭제 횟수가 임계치를 넘는가를 판단하는 제 72 단계(S72); 상기 제 72 단계(S72)에서 판단한 결과 상기 이동 유니트(TU)의 삭제 횟수가 임계치를 넘으면 삭제 횟수가 최소인 이동 유니트(TU)를 선택하는 제 74 단계(S73); 상기 제 72 단계(S72)에서 판단한 결과 상기 이동 유니트(TU)의 삭제 횟수가 임계치를 넘지 않거나 또는 상기 제 73 단계(S73)에서 삭제 횟수가 최소인 이동 유니트(TU)가 선택되면, 상기 재생할 유니트(EU)의 상태를 "이동중"으로 갱신하는 제 74 단계(S74); 상기 제 74 단계(S74)에서 삭제 횟수가 임계치를 넘지 않는 이동 유니트(TU) 또는 삭제 횟수가 최소인 이동 유니트(TU)의 상태를 "복사중"으로 갱신하는 제 75 단계(S75); 상기 재생할 유니트(EU)에서 "기록중" "기록됨" 상태인 블록과 맵 블록 항목, 삭제 횟수를 상기 이동 유니트(TU)로 복사하는 제 76 단계(S76); 상기 이동 유니트(TU)의 상태를 "유효" 로 갱신하는 제 77 단계(S77); 상기 재생할 유니트(EU)를 삭제하는 제 78 단계(S78); 상기 이동 유니트(TU)에 복사된 유니트의 삭제 횟수를 읽어서 증가시킨 후 상기 삭제된 유니트에 기록하는 제 79 단계(S79); 사상 정보 등 플래시 메모리내 자료 구조를 갱신하는 제 80 단계(S80)로 재생 연산을 제어한다.

도 10 은 도 8 의 갱신 연산 수행중 에러 발생시 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 회복 동작의 상태도이다.

도 10 에 도시된 바와 같이 본 발명의 방법은, 어느 한 데이터 블록(X<sub>old</sub>)의 데이터를 삭제중(deleting)일 때, a) 예비 블록(X<sub>new</sub>)이 할당되지 않은 상태(free)에서 에러가 발생하면, 동일 유니트내의 어느 한 예비 블록(Y<sub>new</sub>)을 할당하여 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 할당된 예비 블록(Y<sub>new</sub>)에 기록하고, b) 예비 블록(X<sub>new</sub>)이 할당된 상태(allocated)에서 에러가 발생

하면, 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 할당된 예비 블록(X<sub>new</sub>)에 기록하고, c) 예비 블록(X<sub>new</sub>)에 갱신 데이터를 기록중인 상태(writing)에서 에러가 발생하면, 상기 예비 블록(X<sub>new</sub>)에 기록된 갱신 데이터를 삭제하는 한편 동일 유니트내의 어느 한 예비 블록(Z<sub>new</sub>)을 재할당하여 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 재할당된 예비 블록(Z<sub>new</sub>)에 기록하여, 에러 발생시 회복 동작을 제어한다.

도 11 은 도 9 의 갱신 연산 수행중 에러 발생시 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 회복 동작의 상태도이다.

도 11 에 도시된 바와 같이 본 발명의 방법은, 재생할 유니트(EU)의 데이터를 전송중(transfer)일 때, a) 이동 유니트(TU)에 데이터가 복사중이 아닌 상태(free)에서 에러가 발생하면, 상기 재생할 유니트(EU)의 재생 연산을 재수행하고, b) 이동 유니트(TU)가 복사중인 상태(copying)에서 에러가 발생하면, 상기 이동 유니트(TU)에 복사된 데이터를 삭제하는 한편 상기 재생할 유니트(EU)의 재생 연산을 재수행하여, 에러 발생시 회복 동작을 제어한다.

상기와 같이 구성된 본 발명의 동작 및 효과를 도 2 를 참조하여 플래시 메모리의 유니트 구조, 사상 동작, 블록 할당 동작, 독출 연산, 삭제 연산, 갱신 연산, 재생 연산, 에러 발생시 회복 동작으로 나누어 자세히 살펴보도록 한다.

도 2 는 본 발명이 적용되는 플래시 메모리와 그 주변 회로를 도시한 블록도이다.

## 1. 플래시 메모리의 유니트 구조

본 발명의 방법에 적용되는 플래시 메모리의 유니트 구조는 대체로 FTL, DMS 등과 유사하지만 이들과는 사상 방법이 다르기 때문에 그에 맞는 저장 구조와 알고리즘을 사용한다.

일반적으로 플래시 메모리는 복수개의 유니트로 구성되어 있으며, 유니트는 복수개의 블록으로 구성되어 있다.

각 유니트에 물리적 유니트 번호(PUN : Physical Unit Number)와 논리적 유니트 번호(LUN : Logical Unit Number)가 부여되는데, 물리적 유니트 번호(PUN)는 유니트의 물리적 순서에 따라 매겨지며, 논리적 유니트 번호(LUN)는 유니트의 논리적 순서를 나타내는 번호이다.

또한 각 블록에 물리적 블록 번호(PBN : Physical Block Number)와 논리적 블록 번호(LBN : Logical Block Number)가 부여되는데, 물리적 블록 번호(PBN)는 블록의 물리적 순서에 따라 매겨지며, 논리적 블록 번호(LBN)는 블록의 논리적 순서를 나타내는 번호이다.

도 3 은 본 발명에 따른 플래시 메모리와 그 제어 방법을 설명하기 위한 도면으로, (a)는 LBN-to-LUN 맵 테이블의 구성도, (b)는 LUN-to-PUN 맵 테이블의 구성도, (c)(d)는 플래시 메모리내 유니트의 구성도이다.

도 3 (c)(d)에 도시된 바와 같이 유니트의 첫번째 블록에는 플래시 메모리 전체에 대한 정보(예컨대 블록의 크기 정보)와 그 유니트의 관리에 필요한 정보(예컨대 논리적 유니트 번호(LUN), 마모도)가 기록된다. 이 첫번째 블록을 헤더(EUH : Eraser Unit Header) 블록이라 한다.

유니트에서 두번째 블록은 맵(BAM : Block Allocation Map) 블록으로, 상기 맵 블록에는 그 유니트에 속한 블록들에 대한 정보(논리적 블록 번호(LBN), 블록의 상태 정보, 갱신된 블록의 위치 정보 등)가 기록된다.

상기 논리적 블록 번호(LBN)는 사용자가 생각하는 블록의 주소이다. 따라서 상기 맵 블록은 논리적 블록 번호(LBN)와 그 블록이 실제로 기록된 물리적 블록번호(PBN)간의 사상을 나타낸다. 블록의 데이터가 갱신되면 상기 맵 블록에서 해당 항목의 상태 정보만 바꾸고 갱신된 블록이 저장된 위치를 기록한다. 이값을 따라가면서 블록의 최종 위치를 알아낼 수 있다.

이때, 좀더 효율적으로 플래시 메모리(1)를 제어하기 위해 상기 각 유니트내 맵 블록의 사본(일부 혹은 전부)을 RAM(3)과 같은 저장수단에 저장해두므로써, 변경된 위치 정보를 따라가지 않고도 즉시 파악하게 할 수 있다.

이때, 상기 유니트에 할당될 수 있는 블록의 수는 그 유니트에 기록될 수 있는 최대 블록 수보다 작게 유지한다.

예를들어 도 3(c)(d)에 도시된 바와 같이 한 유니트에 6개의 블록이 있다면, 한 개의 블록은 헤더 블록으로, 한 개의 블록은 맵 블록으로, 세 개의 블록은 데이터 블록으로, 나머지 한 개의 블록은 예비 블록으로 나누어 사용하므로써, 상기 데이터 블록이 갱신되거나 삭제된 후 재기록될 때 상기 예비 블록을 이용할 수 있다.

## 2. 사상 동작

본 발명의 방법에 의해 제어되는 사상 동작은 은 고정 사상과 임의 사상을 혼합한 일종의 혼합 사상이다.

도 1(c)에 도시된 바와 같이, 혼합 사상은 논리적 블록 번호(LBN)를 특정한 논리적 유니트 번호(LUN)에 할당하고 논리적 유니트내에서 논리적 블록 번호(LBN)를 임의로 할당한다. 논리적 블록 번호(LBN)를 논리적 유니트 번호에 할당하는 방식은 마치 해시 함수를 이용하여 키값을 버킷에 할당하는 것과 유사하다. "Search(Hash(LBN))=PBN".

도 4 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 사상 동작의 흐름도로서, 도 1(c)와 도 3 및 도 4 를 참조하여 살펴보면 다음과 같다.

먼저, 중앙처리장치(1)가 연산을 수행하기 위해 플래시 메모리(2)의 어느 한 블록에 접근하려면 우선 그

블럭이 속하는 논리적 유니트 번호(LUN)를 탐색해야 한다(S11). 이때 논리적 블럭 번호(LBN)가 속하는 논리적 유니트 번호(LUN)를 탐색하기 위해서 도 3(a)에 도시된 바와 같은 "LBN-to-LUN" 맵 테이블을 이용할 수 있다.

상기 중앙처리장치(1)가 논리적 유니트 번호(LUN)를 결정한 뒤에는 도 3(b)에 도시된 바와 같은 "LUN-to-PUN" 맵 테이블을 이용하여 논리적 유니트에 대한 물리적 유니트를 결정한다(S12).

그리고 나서 상기 중앙처리장치(1)는 도 3(c)(d)에 도시된 바와 같은 물리적 유니트의 맵 블럭에서 상기 논리적 블럭에 대한 물리적 블럭의 위치를 탐색한다(S13).

상기 "LBN-to-LUN" 맵 테이블과 "LUN-to-PUN" 맵 테이블은 RAM(3)상에 위치하며, 삭제 연산이나 갱신 연산 혹은 재생 연산으로 인해 사상 관계가 변하면 상기 맵 테이블도 갱신되어 동일한 논리적 주소로 계속 접근할 수 있다.

이같은 혼합 사상은 고정 사상과 같이 논리적 블럭이 물리적 블럭에 일정하게 사상되지는 않으며 복잡한 사상 정보를 유지하기 위한 공간과 시간이 임의의 사상만큼 필요하지는 않으며, 어떤 한 유니트에 속한 블럭들이 집중적으로 갱신되는 경우 삭제 연산이 빈번히 발생하지 않기 때문에 데이터 기록 및 갱신의 효율을 높일 수 있다.

이중 बैंक 플래시 메모리(2)에서는 여러 크기의 유니트를 사용하는 경우가 있는데, 이런 경우 유니트의 크기도 서로 다르기 때문에, 유니트를 논리적 블럭이 할당되는 버킷으로 사용하는 본 발명에 의한 사상 동작에서 이같은 비대칭 구조의 플래시 메모리를 지원하기 위해서는 서로 다른 크기의 버킷을 효율적으로 사용하는 사상 방법이 마련되어야 하는 바, 도 1 (c)에 도시된 바와 같이 논리적 블럭 번호(LBN)를 버킷번호(BN)으로 사상(LBN-to-BN)하는 한편 버킷 번호(BN)를 논리적 유니트 번호(LUN)로 사상(BN-to-LUN)함으로써, 마치 크기가 다른 버킷이 사용되는 것 처럼 처리한다.

### 3. 블럭 할당 동작

도 5 는 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 블럭 할당 동작의 흐름도이다.

도 5 에 도시된 바와 같이 중앙처리장치(1)가 먼저 플래시 메모리(2)내 유니트의 맵 블럭을 검사하여 상기 유니트에 예비 블럭이 있는가를 판단하여(S21), 상기 유니트에 예비 블럭이 없으면 상기 유니트에 대해 재생연산을 수행한다(S22).

이에 따라 상기 중앙처리장치(1)가 상기 재생된 유니트의 맵 블럭을 검사하여 상기 재생된 유니트에 예비 블럭이 있는가를 판단하여(S23), 상기 제 21 단계(S21)에서 판단한 결과 상기 유니트에 예비 블럭이 있거나 또는 상기 제 23 단계(S23)에서 판단한 결과 상기 재생된 유니트에 예비 블럭이 있으면 맵 블럭에 데이터 블럭의 논리적 블럭 번호(LBN)와 상태를 기록하고(S24), 상기 재생된 유니트에 예비 블럭이 없으면 에러 처리한다(S25).

이때 상기 유니트에 할당될 수 있는 데이터 블럭의 수는 그 유니트에 기록될 수 있는 최대 블럭 수보다 작게 유지한다.

예를들어 한 유니트에 최대 10개의 블럭이 기록될 수 있다면, 실제로 7개의 블럭만 유니트에 할당한다.

나머지 3개 만큼의 빈 공간은 기존의 블럭이 갱신되거나 삭제된 후 재기록될 때 이용될 수 있도록 예비 블럭으로 남겨둔다.

데이터 블럭에 대한 예비 블럭의 비율은 플래시 메모리의 응용 요구에 따라 결정될 수 있는 바, 갱신 연산이 자주 이루어지는 유니트에는 예비 블럭의 비율을 높여서 재생 연산의 수행 빈도를 낮추므로써, 전체적으로 연산의 수행 속도를 증가시킨다.

독출 연산 전용인 유니트라면 예비 블럭을 따로 두지 않으므로써, 공간 사용률을 높일 수 있다.

### 4. 독출 연산

도 6 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 독출 연산의 흐름도이다.

도 3 과 도 6 을 참조하여 독출 연산을 살펴보면, 중앙처리장치(1)가 먼저 혼합 사상에 의해 해당 데이터 블럭이 속한 논리적 유니트 번호(LUN)와 물리적 유니트 번호(PUN)를 탐색하고, 해당 물리적 유니트의 맵 블럭을 통해 해당 데이터 블럭의 물리적 위치를 결정한다(S31).

이때 상기 중앙처리장치(1)는 상기 유니트내에 해당 데이터 블럭이 존재하는가를 판단하여(S32), 해당 데이터 블럭이 존재하면 상기 해당 데이터 블럭에서 데이터를 독출하고(S33), 해당 데이터 블럭이 존재하지 않으면 에러 처리한다(S34).

갱신된 블럭인 경우에는 맵 블럭에서 갱신된 블럭의 위치를 따라가면 블럭의 위치를 찾을 수 있다.

### 5. 삭제 연산

도 7 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 삭제 연산의 흐름도이다.

도 7 에 도시된 바와 같이 중앙처리장치(1)는 먼저 혼합 사상에 의해 해당 데이터 블럭이 속한 논리적 유니트 번호(LUN)와 물리적 유니트 번호(PUN)를 탐색하여, 해당 물리적 유니트의 맵 블럭을 통해 해당 데이터 블럭의 물리적 위치를 결정한 후(S41), 상기 유니트의 맵 블럭에 기록된 해당 데이터 블럭의 상태를 "삭제중"으로 갱신한다(S42).

즉, 블럭의 데이터를 삭제할 때는 그 블럭이 속한 유니트를 삭제하지 않고 블럭의 상태 정보만 갱신한다.

### 6. 갱신 연산

도 8 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 갱신 연산의 흐름도이고, 도 12 은 도 8 의 갱신 연산을 설명하기 위한 도면으로 (a)는 LBN-to-LUN 맵 테이블의 구성도, (b)는 LUN-to-PUN 맵 테이블의 구성도, (c)(d)는 플래시 메모리의 유닛의 구성도이다.

도 8 과 도 12 을 참조하여 본 발명에 따른 갱신 연산을 살펴보면, 중앙처리장치(1)는 먼저 혼합 사상에 의해 해당 데이터 블록이 속한 논리적 유닛 번호(LUN)와 물리적 유닛 번호(PUN)를 탐색하여, 해당 물리적 유닛의 맵 블록을 통해 해당 데이터 블록의 물리적 위치를 결정한다(S51).

상기 중앙처리장치(1)는 상기 유닛내에 해당 데이터 블록이 존재하는가를 판단하여(S52), 해당 데이터 블록이 존재하면 맵 블록을 검사하여 해당 데이터 블록의 상태가 "할당됨"인가를 판단한다(S53).

상기 제 53 단계(S53)에서 판단한 결과 해당 블록이 "할당됨" 상태이면, 상기 중앙처리장치(1)는 해당 데이터 블록 상태를 "기록중"으로 갱신 한 후(S54), 해당 데이터 블록에 데이터를 기록하고(S55), 다시 해당 데이터 블록 상태를 "기록됨"으로 갱신한다(S56).

상기 제 53 단계(S53)에서 판단한 결과 해당 데이터 블록이 "할당됨" 상태이면 상기 중앙처리장치(1)는 블록의 상태를 "삭제중"으로 갱신하고(S57), 상기 유닛내에서 예비 블록을 탐색하여 할당한다(S58).

이에 따라 중앙처리장치(1)는 새롭게 할당된 블록의 상태를 "기록중"으로 갱신하고(S59), 상기 새롭게 할당된 블록의 위치 정보를 상기 맵 블록에 기록한 후(S60), 상기 새롭게 할당된 블록에 데이터를 기록하고(S61), 상기 새롭게 할당된 블록의 상태를 "기록됨"으로 갱신한 후(S62), 상기 해당 데이터 블록의 상태를 "삭제됨"으로 갱신한다(S63).

이때 상기 제 52 단계(S52)에서 판단한 결과 해당 데이터 블록이 존재하지 않으면 상기 중앙처리장치(1)는 에러 처리한다(S64).

즉, 블록의 데이터를 갱신할 때는, 유닛내의 예비 블록이 있으면 그곳에 새로운 데이터를 기록하고 사상 정보를 갱신하며 이전 블록의 상태를 갱신한다.

예를들어 도 12(c) 에 도시된 바와 같이 블록 1의 데이터를 갱신할 경우, 상기 중앙처리장치(1)는 예비 블록을 새로운 블록 4로 할당하고 새로운 데이터를 상기 새로운 블록 4에 기록함으로써, 물리적 유닛 1에는 더이상 예비 블록이 없게 된다. 이때 이전 블록 1의 데이터는 그대로 두고 블록1의 상태 정보만 "삭제됨"으로 표시한다.

## 7. 재생 연산

유닛에 할당된 블록이 갱신되거나 삭제되는 일이 계속 발생하면 예비 블록이 점점 줄어들어서 더 이상 갱신과 삭제가 수행될 수 없게 된다.

임의 사상이라면 다른 유닛으로 블록을 이동시키면 되지만 본 발명에서는 블록이 할당되는 논리적 유닛이 일정하기 때문에, 유닛을 삭제하여 예비 블록을 다시 확보해야 한다.

즉, 재생할 유닛에서 유효 블록들을 이동 유닛으로 옮기고 이전의 유닛을 삭제하여 다시 사용 가능하도록 재생시킨다.

이때 이동 뒤에는 삭제된 유닛이 이동 유닛이 된다.

플래시 메모리장치에는 적어도 하나 이상의 이동 유닛이 항상 존재하는 데, 유닛의 크기가 다를 경우에는 적어도 그 종류만큼의 이동 유닛이 있어야 한다.

예를들어 8KB와 64KB의 유닛으로 구성된 플래시 메모리라면 8KB와 64KB 이동 유닛이 각각 하나 이상 존재해야 한다.

이동은 크기가 같은 유닛 사이에만 발생한다.

따라서 이동 후에도 BN-to-LUN 맵 테이블에서 논리적 유닛 번호(LUN)는 갱신되지 않으며 LUN-to-PUN 맵 테이블에서 물리적 유닛 번호(PUN)만 갱신된다.

크기가 다른 유닛 사이에서 이동이 일어나게 하려면 BN-to-LUN 사상을 위한 별도의 저장 공간이 필요하며, 유닛 내에서 버킷의 경계를 유지하기 위한 저장 공간과 처리 시간도 필요하다.

본 발명의 방법에서는 논리적 블록이 일정한 논리적 유닛으로 사상되기 때문에 어떤 특정 블록이 자주 갱신되는 경우 그것이 속한 논리적 유닛의 삭제 횟수가 급속히 증가할 수 있다.

이러한 단점을 파일 팩터(Fill Factor)와 마모도 평준화 기법을 사용하여 보완한다.

즉 버킷의 크기를 유닛의 크기보다 작게 유지함으로써 갱신 연산이 어느정도 일어날 때 까지는 이동이 발생하지 않게 할 수 있다.

그러나 그렇게 해도 어떤 경우에는 두개의 유닛의 삭제 횟수만 집중적으로 증가할 수 있다.

따라서 헤더 블록에 삭제 횟수를 기록하여 그것이 규정된 값을 넘으면 삭제 횟수가 적은 다른 유닛과 데이터를 교환한다.

도 9 은 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 재생 연산의 흐름도이고, 도 13 은 도 9 의 재생 연산을 설명하기 위한 도면으로 (a)는 LBN-to-LUN 맵 테이블의 구성도, (b)는 LUN-to-PUN 맵 테이블의 구성도, (c)(d)는 플래시 메모리내 유닛의 구성도이다.

도 9 와 도 13 을 참조하여 본 발명에 따른 재생 연산을 살펴보면, 먼저 중앙처리장치(1)가 재생할 유닛(EU)와 이동 유닛(TU)를 결정하면(S71), 상기 이동 유닛(TU)의 헤더 블록에 기록된 삭제 횟수를 검출하여 상기 삭제 횟수가 임계치를 넘는가를 판단한다(S72).



상기 제 72 단계(S72)에서 판단한 결과 상기 이동 유닛(TU)의 삭제 횟수가 임계치를 넘으면 상기 중앙처리장치(1)는 삭제 횟수가 최소인 이동 유닛(TU)를 선택한다(S73).

상기 제 72 단계(S72)에서 판단한 결과 상기 이동 유닛(TU)의 삭제 횟수가 임계치를 넘지 않거나 또는 상기 제 73 단계(S73)에서 삭제 횟수가 최소인 이동 유닛(TU)가 선택되면, 상기 중앙처리장치(1)는 상기 재생할 유닛(EU)의 상태를 "이동중"으로 갱신하고(S74), 상기 삭제 횟수가 임계치를 넘지 않는 이동 유닛(TU) 또는 삭제 횟수가 최소인 이동 유닛(TU)의 상태를 "복사중"으로 갱신한다(S75).

상기 중앙처리장치(1)는 상기 재생할 유닛(EU)에서 "기록중" "기록됨" 상태인 블록과 맵 블록 항목, 삭제 횟수를 상기 이동 유닛(TU)로 복사하고(S76), 상기 이동 유닛(TU)의 상태를 "유효"로 갱신한 후(S77), 상기 재생할 유닛(EU)를 삭제한다(S78).

상기 과정이 완료되면 중앙처리장치(1)는 상기 이동 유닛(TU)의 헤더 블록에 복사된 삭제 횟수를 읽어서 증가시킨 후 상기 삭제된 유닛의 헤더 블록에 기록하고(S79), 사상 정보 등 플래시 메모리내 자료 구조를 갱신한다(S80).

이때 상기 제 72 단계(S72)에서 판단한 결과 삭제 횟수가 임계치를 넘는 이동 유닛(TU)의 번호를 RAM(3)에 따로 기록해두었다가 상기와 같은 재생 동작이 완료되고 나면 상기 RAM(3)에 기록된 이동 유닛의 번호를 독출하여 상기 이동 유닛에 대해 재생 연산을 수행한다.

예를들어 도 13(c)(d)에 도시된 바와 같이 물리적 유닛 번호가 1일 유닛에 예비 블록이 존재하지 않으면 상기 중앙처리장치(1)는 재생 연산을 수행하여, 재생할 유닛에서 유효 블록들을 물리적 유닛 번호가 3인 이동 유닛으로 옮기고 물리적 유닛 번호가 1인 유닛의 데이터를 모두 삭제하여 예비 블록을 다시 확보해야 한다.

재생 연산 수행시 상기 중앙처리장치(1)는 이동 유닛의 헤더 블록에서 삭제 횟수를 독출하여 삭제 횟수에 따라 재생 여부를 결정할 수도 있지만, 시스템 초기화시 결정하여 번역 변수로 유지할 수 있다.

또한, 상기 중앙처리장치(1)는 마모도 평준화를 위하여 마모도 순위를 검색해야 하는데, 이를 위해 메모리내의 큐에 자료 구조를 구성할 수 있다.

또한, 상기 중앙처리장치(1)는 상기 블록 또는 유닛의 상태를 플래시 메모리의 특성과 알고리즘상의 순서에 맞도록 미리 설정할 수 있다.

예를 들어 독출, 기록, 삭제, 갱신 연산에서 블록의 상태가 미정(FF)→할당됨(8F)→기록중(4F)→기록됨(2F)→삭제중(0F)→삭제됨(00)의 순서로 진행되고, 재생 연산에서 유닛의 상태가 미정(FF)→복사중(8F)→유효(4F)→이동중(2F)의 순서로 진행되므로, 블록 또는 유닛의 상태가 상기 순서에 따라 갱신되도록 미리 설정해둔다.

## 8. 에러 회복

도 10 은 도 8 의 갱신 연산 수행중 에러 발생시 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 회복 동작의 상태도이고, 도 11 은 도 9 의 재생 연산 수행중 에러 발생시 본 발명에 따른 플래시 메모리의 제어 방법에 의해 제어되는 회복 동작의 상태도이다.

즉, 도 10 은 블록을 갱신할 때 이전 블록(X<sub>old</sub>)과 새로운 블록(X<sub>new</sub>)의 상태 정보가 시간에 따라 갱신되는 단계를 보여준다.

아래로 향한 굵은 화살표는 시간축을 나타내며 사각형 안에 든 값은 해당 시점에서의 블록 상태를 나타내고, 가로로 그어진 선은 고장 발생 시점을 나타낸다. 그림 좌우의 굵어진 화살표와 레이블은 회복 단계에서 취할 연산과 회복 후 그 블록이 돌아갈 상태를 나타낸다.

이에 따라 어느 한 데이터 블록(X<sub>old</sub>)의 데이터를 삭제중(deleting)일 때, a) 예비 블록(X<sub>new</sub>)이 할당되지 않은 상태(free)에서 에러가 발생하면, 동일 유닛내의 어느 한 예비 블록(Y<sub>new</sub>)을 할당하여 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 할당된 예비 블록(Y<sub>new</sub>)에 기록하고, b) 예비 블록(X<sub>new</sub>)이 할당된 상태(allocated)에서 에러가 발생하면, 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 할당된 예비 블록(X<sub>new</sub>)에 기록하고, c) 예비 블록(X<sub>new</sub>)에 갱신 데이터를 기록중인 상태(writing)에서 에러가 발생하면, 상기 예비 블록(X<sub>new</sub>)에 기록된 갱신 데이터를 삭제하는 한편 동일 유닛내의 어느 한 예비 블록(Z<sub>new</sub>)을 재할당하여 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 재할당된 예비 블록(Z<sub>new</sub>)에 기록하여, 에러 발생시 회복 동작을 제어한다.

예를들어 데이터 블록(X<sub>old</sub>)의 상태가 "삭제중(deleting)"이고 예비 블록(X<sub>new</sub>)의 상태가 "할당됨(allocated)"일 때 고장이 발생하면, 재시작시 상기 데이터 블록(X<sub>old</sub>)에서 삭제중인 데이터를 상기 할당된 예비 블록(X<sub>new</sub>)에 기록하면서 나머지 과정을 계속 진행한다.

블록의 상태가 가질 수 있는 값은 도 10 의 상태도에 모두 표현되어 있으며, 회복 과정을 수행하는 도중에 다시 고장이 발생하여도 상태도의 순서에 따라 다시 회복할 수 있다.

도 13 은 유닛을 재생시킬 때 재생되는 유닛(EU)와 비어있는 유닛(TU)의 상태 변화도이다.

이에 따라 재생할 유닛(EU)의 데이터를 전송중(transfer)일 때, a) 이동 유닛(TU)에 데이터가 복사중이 아닌 상태(free)에서 에러가 발생하면, 상기 재생할 유닛(EU)의 재생 연산을 재수행하고, b) 이동 유닛(TU)가 복사중인 상태(copying)에서 에러가 발생하면, 상기 이동 유닛(TU)에 복사된 데이터를 삭제하는 한편 상기 재생할 유닛(EU)의 재생 연산을 재수행하여, 에러 발생시 회복 동작을 제어한다.

예를들어 재생할 유닛(EU)의 상태가 "전송중(transfer)"이고 이동 유닛(TU)의 상태가 "복사중(copying)"일 때 고장이 발생하면, 재시작시 상기 이동 유닛(TU)에 복사된 데이터를 삭제하는 한편 상기 재생할 유닛(EU)의 재생 연산을 재수행한다.

따라서 재생이 완전히 수행되거나 아니면 전혀 수행되지 않은 상태만 존재하게 된다.

### **발명의 효과**

데이터 블록의 데이터를 갱신하고자 할 경우 먼저 갱신할 데이터를 동일 유니트내의 비어있는 예비 블록에 기록한 후, 맵 블록의 사상정보를 갱신하고 이전 블록의 상태정보를 갱신함으로써, 사용자가 동일한 주소를 이용하여 데이터에 접근할 수 있어, 블록을 갱신할 때마다 유니트를 지우지 않아도 되기 때문에, 삭제 횟수를 감소시켜서 데이터 기록 및 갱신의 효율을 높인다는 데 그 효과가 있다. 또한 플래시 메모리 사용중 에러가 발생하더라도 그 회복이 용이하며, 사상의 임의성이 감소하기 때문에 발생하는 집중적인 삭제 연산을 해결한다는 데 그 효과가 있다.

### **(57) 청구의 범위**

#### **청구항 1**

데이터를 기록하기 위한 복수개의 데이터 블록 ;

상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록 ;

데이터가 기록된 실제 위치를 지정하는 맵 블록 ;

플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성된 것을 특징으로 하는 플래시 메모리.

#### **청구항 2**

제 1 항에 있어서,

상기 유니트가 삭제 연산의 기본 단위로 사용되는 것을 특징으로 하는 플래시 메모리.

#### **청구항 3**

제 1 항에 있어서,

상기 헤더 블록에 논리적 유니트 번호, 마모도, 블록의 크기에 대한 정보를 기록하는 것을 특징으로 하는 플래시 메모리.

#### **청구항 4**

제 1 항에 있어서

상기 맵 블록에 논리적 블록 번호, 블록의 상태 정보, 갱신된 블록의 위치 정보를 기록하는 것을 특징으로 하는 플래시 메모리.

#### **청구항 5**

데이터를 기록하기 위한 복수개의 데이터 블록 ; 상기 데이터 블록의 갱신 데이터를 기록하기 위해 비워 둔 복수개의 예비 블록 ; 데이터가 기록된 실제 위치를 지정하는 맵 블록 ; 플래시 메모리에 대한 정보와 유니트에 대한 정보를 기록하는 헤더 블록으로 이루어진 유니트를 포함하여 구성된 플래시 메모리에 있어서,

논리적 블록이 속하는 논리적 유니트를 탐색하는 과정 ;

상기 논리적 유니트에 대한 물리적 유니트를 탐색하는 과정 ;

상기 물리적 유니트의 맵 블록에서 상기 논리적 블록에 대한 물리적 블록의 위치를 탐색하는 과정으로 사상 동작을 제어하는 것을 특징으로 하는 플래시 메모리의 제어방법.

#### **청구항 6**

제 5 항에 있어서,

상기 유니트의 맵 블록을 검사하여 상기 유니트에 예비 블록이 있는가를 판단하는 제 21 단계 ;

상기 제 21 단계에서 판단한 결과 상기 유니트에 예비 블록이 없으면 상기 유니트에 대해 재생연산을 수행하는 제 22 단계 ;

상기 재생된 유니트의 맵 블록을 검사하여 상기 재생된 유니트에 예비 블록이 있는가를 판단하는 제 23 단계 ;

상기 제 21 단계에서 판단한 결과 상기 유니트에 예비 블록이 있거나 또는 상기 제 23 단계에서 판단한 결과 상기 재생된 유니트에 예비 블록이 있으면, 상기 맵 블록에 데이터 블록의 논리적 블록 번호와 상태를 기록하는 제 24 단계 ;

상기 제 23 단계에서 판단한 결과 상기 재생된 유니트에 예비 블록이 없으면 에러 처리하는 제 25 단계로 할당 연산을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

#### **청구항 7**

제 5 항에 있어서,

해당 데이터 블록이 속한 유닛을 탐색하는 제 31 단계 ;

상기 유닛내에 해당 데이터 블록이 존재하는가를 판단하는 제 32 단계 ;

상기 제 32 단계에서 판단한 결과 해당 데이터 블록이 존재하면, 상기 해당 데이터 블록에서 데이터를 독출하는 제 33 단계 ;

상기 제 32 단계에서 판단한 결과 해당 데이터 블록이 존재하지 않으면 에러 처리하는 제 34 단계로 독출 연산을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

#### 청구항 8

제 5 항에 있어서,

해당 데이터 블록이 속한 유닛을 탐색하는 제 41 단계 ;

상기 유닛의 맵 블록에 기록된 해당 데이터 블록의 상태를 "삭제중"으로 갱신하는 제 42 단계로 삭제 연산을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

#### 청구항 9

제 5 항에 있어서,

해당 데이터 블록이 속한 유닛을 결정하는 제 51 단계 ;

상기 유닛내에 해당 데이터 블록이 존재하는가를 판단하는 제 52 단계 ;

상기 제 52 단계에서 판단한 결과 해당 데이터 블록이 존재하면, 맵 블록을 검사하여 해당 데이터 블록의 상태가 "활당됨"인가를 판단하는 제 53 단계 ;

상기 제 53 단계에서 판단한 결과 해당 블록이 "활당됨" 상태이면 해당 데이터 블록 상태를 "기록중"으로 갱신하는 제 54 단계 ;

해당 데이터 블록에 데이터를 기록하는 제 55 단계 ;

해당 데이터 블록 상태를 "기록됨"으로 갱신하는 제 56 단계 ;

상기 제 53 단계에서 판단한 결과 해당 데이터 블록이 "활당됨" 상태이면 블록의 상태를 "삭제중"으로 갱신하는 제 57 단계 ;

상기 유닛내에서 예비 블록을 탐색하여 할당하는 제 58 단계 ;

새롭게 할당된 블록의 상태를 "기록중"으로 갱신하는 제 59 단계 ;

상기 새롭게 할당된 블록의 위치 정보를 상기 맵 블록에 기록하는 제 60 단계 ;

상기 새롭게 할당된 블록에 데이터를 기록하는 제 61 단계 ;

상기 새롭게 할당된 블록의 상태를 "기록됨"으로 갱신하는 제 62 단계 ;

상기 해당 데이터 블록의 상태를 "삭제됨"으로 갱신하는 제 63 단계 ;

상기 제 52 단계에서 판단한 결과 해당 데이터 블록이 존재하지 않으면 에러 처리하는 제 64 단계로 갱신 연산을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

#### 청구항 10

제 5 항에 있어서,

재생할 유닛과 이동 유닛을 결정하는 제 71 단계 ;

상기 이동 유닛의 삭제 횟수가 임계치를 넘는가를 판단하는 제 72 단계 ;

상기 제 72 단계에서 판단한 결과 상기 이동 유닛의 삭제 횟수가 임계치를 넘으면 삭제 횟수가 최소인 이동 유닛을 선택하는 제 73 단계 ;

상기 제 72 단계에서 판단한 결과 상기 이동 유닛의 삭제 횟수가 임계치를 넘지 않거나 또는 상기 제 73 단계에서 삭제 횟수가 최소인 이동 유닛이 선택되면, 상기 재생할 유닛의 상태를 "이동중"으로 갱신하는 제 74 단계 ;

상기 74 단계에서 삭제 횟수가 임계치를 넘지 않는 이동 유닛 또는 삭제 횟수가 최소인 이동 유닛의 상태를 "복사중"으로 갱신하는 제 75 단계 ;

상기 재생할 유닛에서 "기록중" "기록됨" 상태인 블록과 맵 블록 항목, 삭제 횟수를 상기 이동 유닛로 복사하는 제 76 단계 ;

상기 이동 유닛의 상태를 "유효" 로 갱신하는 제 77 단계 ;

상기 재생할 유닛을 삭제하는 제 78 단계 ;

상기 이동 유닛에 복사된 유닛의 삭제 횟수를 읽어서 증가시킨 후 상기 삭제된 유닛에 기록하는 제 79 단계 ;

사상 정보 등 플래시 메모리내 자료 구조를 갱신하는 제 80 단계로 재생 연산을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

#### 청구항 11

제 9 항에 있어서,

어느 한 데이터 블록의 데이터를 삭제중일 때,

- a) 예비 블록이 할당되지 않은 상태에서 에러가 발생하면, 동일 유니트내의 어느 한 예비 블록을 할당하여 상기 데이터 블록에서 삭제중인 데이터를 상기 할당된 예비 블록에 기록하고,
- b) 예비 블록이 할당된 상태에서 에러가 발생하면, 상기 데이터 블록에서 삭제중인 데이터를 상기 할당된 예비 블록에 기록하고,
- c) 예비 블록에 데이터를 기록중인 상태에서 에러가 발생하면, 상기 예비 블록에 기록된 데이터를 삭제하는 한편 동일 유니트내의 어느 한 예비 블록을 재할당하여 상기 데이터 블록에서 삭제중인 데이터를 상기 재할당된 예비 블록에 기록하여,

에러 발생시 회복 동작을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

#### 청구항 12

제 10 항에 있어서,

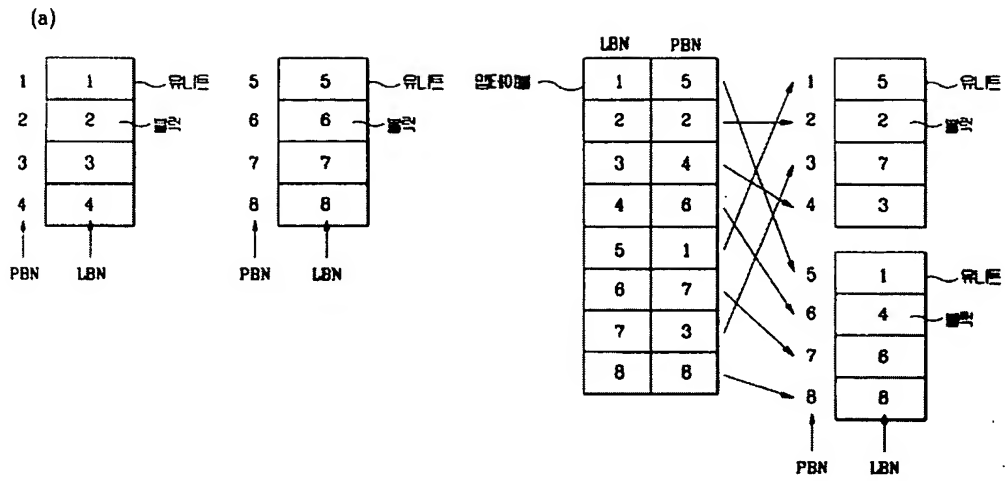
재생할 유니트의 데이터를 전송중일 때,

- a) 이동 유니트에 데이터가 복사중이 아닌 상태에서 에러가 발생하면, 상기 재생할 유니트의 재생 연산을 재수행하고,
- b) 이동 유니트가 복사중인 상태에서 에러가 발생하면, 상기 이동 유니트에 복사된 데이터를 삭제하는 한편 상기 재생할 유니트의 재생 연산을 재수행하여,

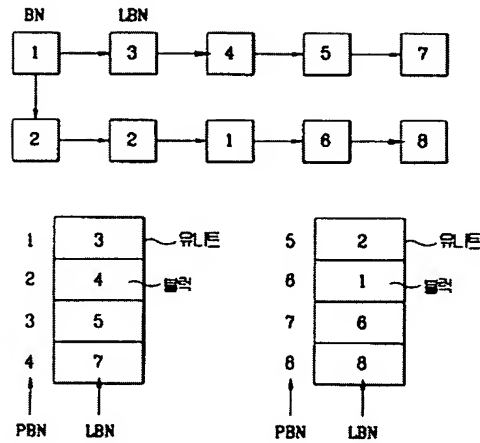
에러 발생시 회복 동작을 제어하는 것을 특징으로 하는 플래시 메모리의 제어 방법.

도면

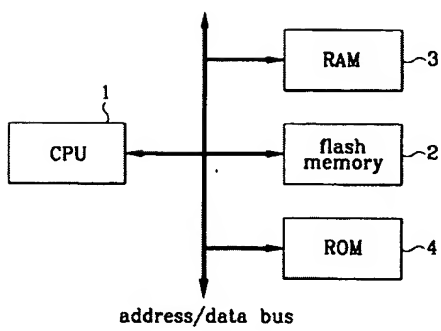
도면1



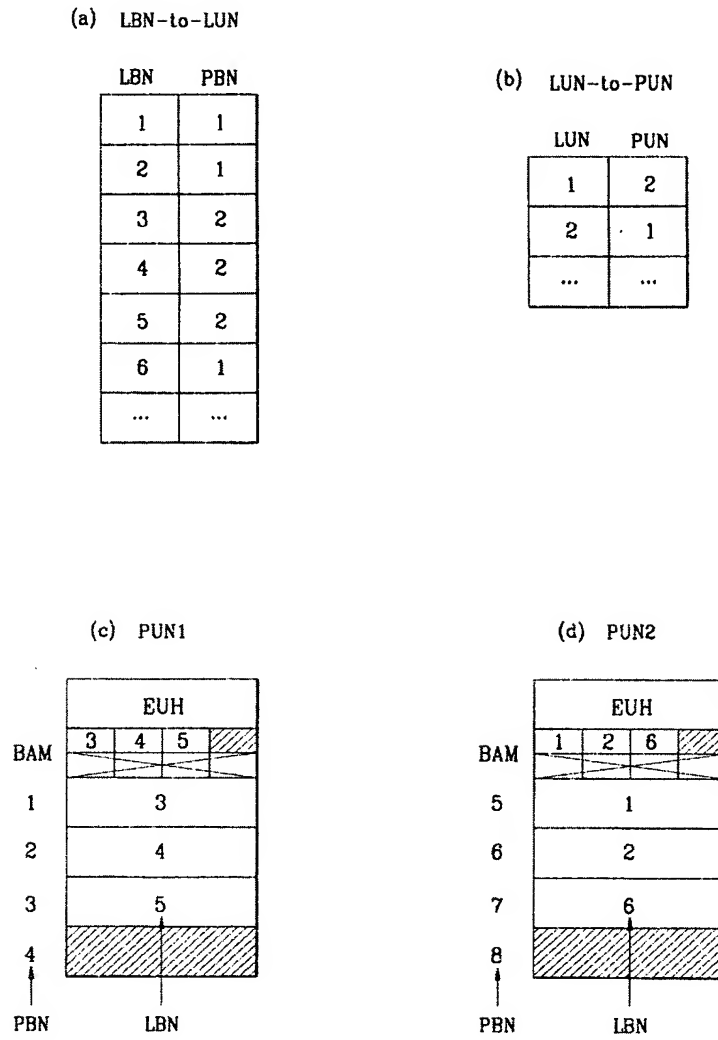
(c)



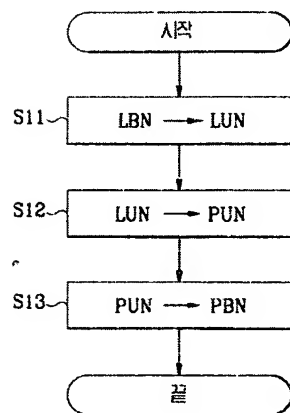
도면2



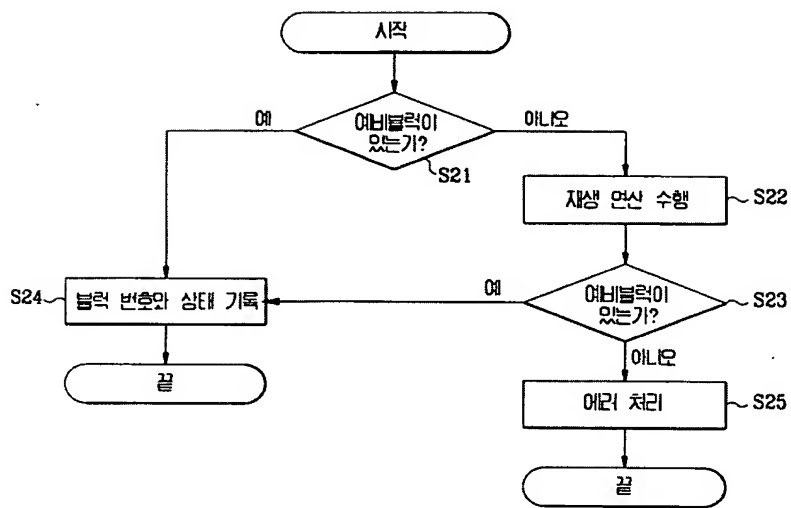
도면3



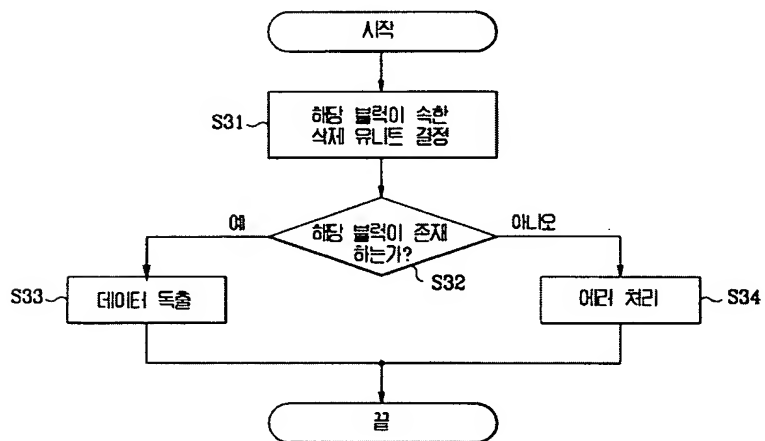
도면4



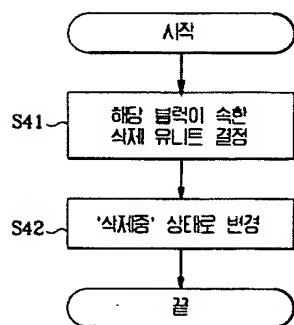
도면5



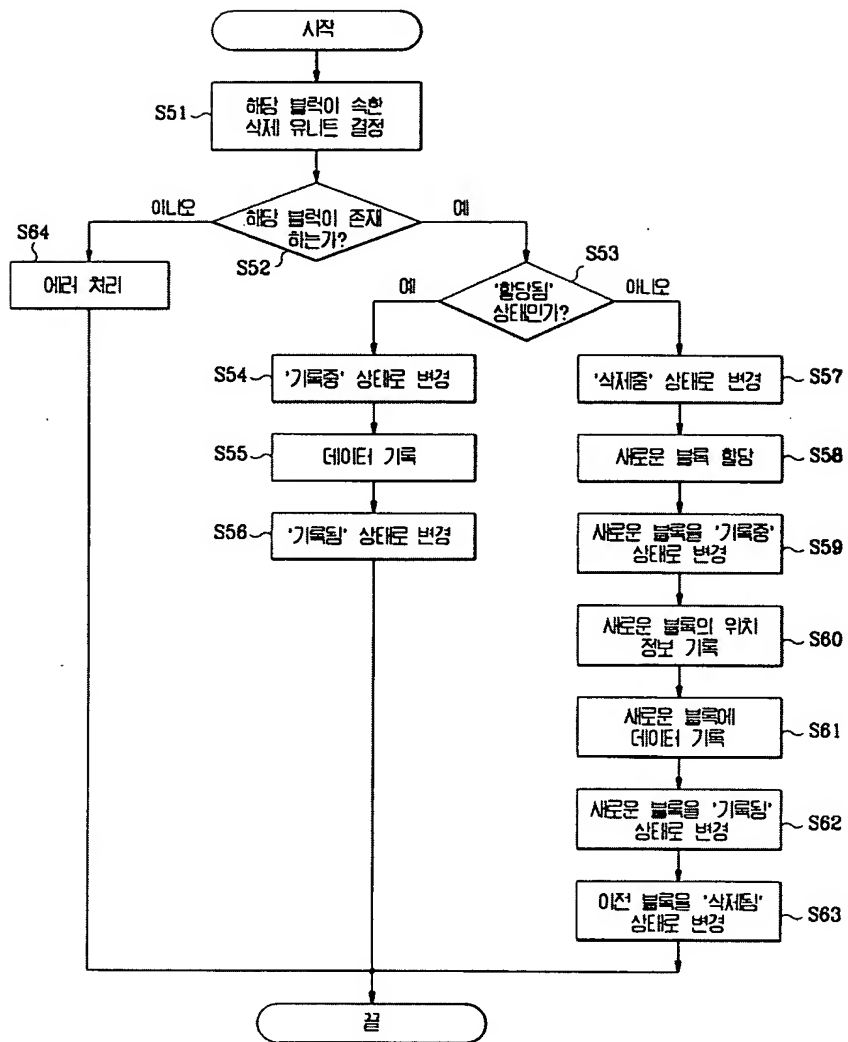
도면6



도면7

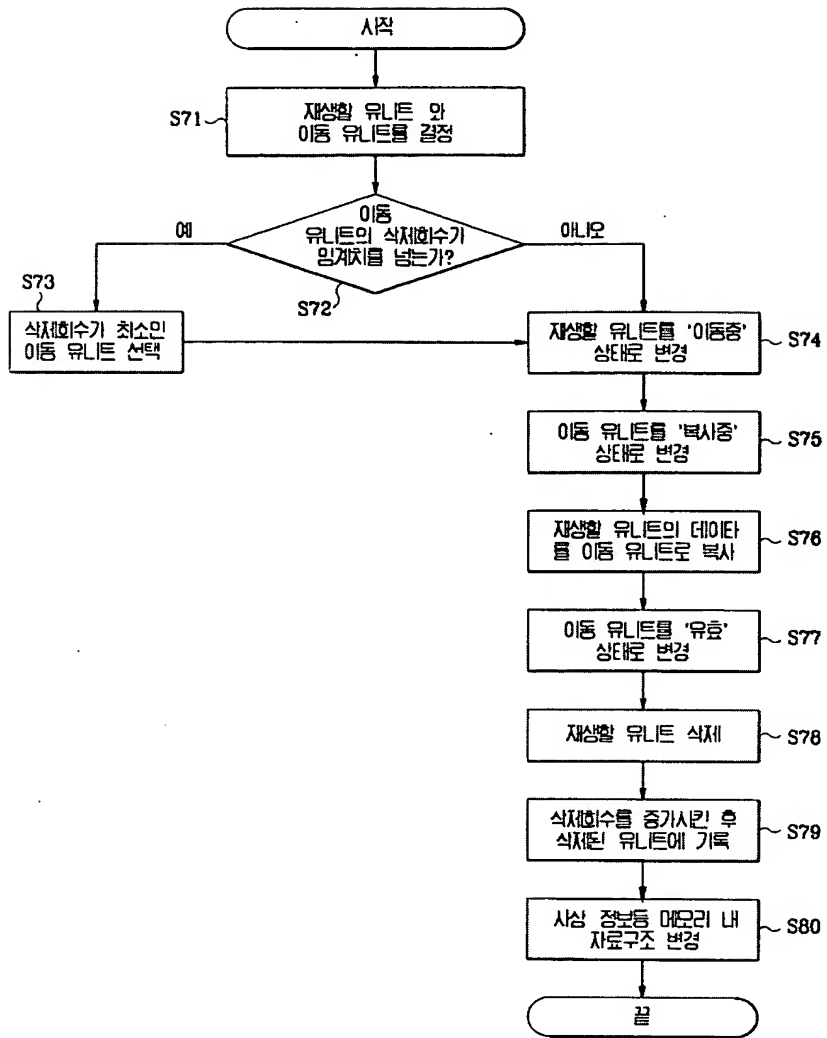


도면 8

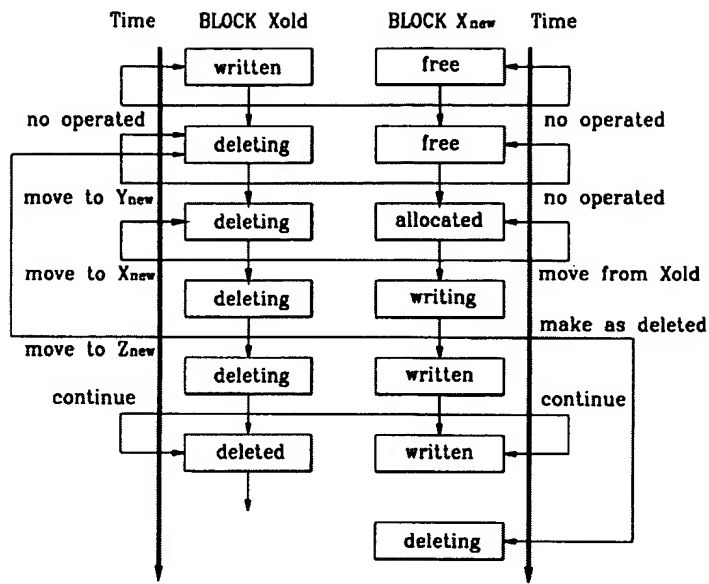




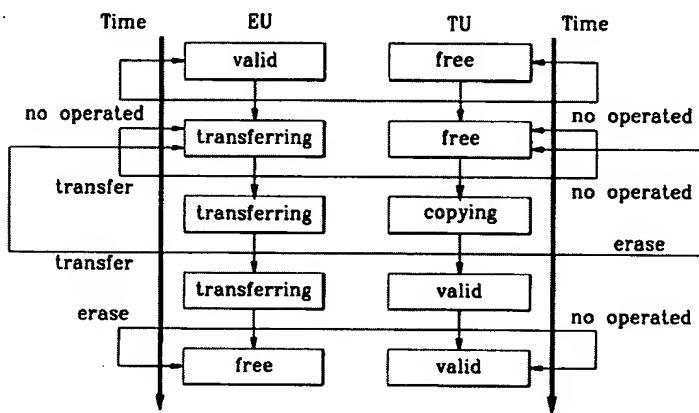
도면9



도면 10



도면 11



도면 12

(a) LBN-to-LUN

LBN	LUN
1	1
2	1
3	2
4	2
5	2
6	1
...	...

(b) LUN-to-PUN

LUN	PUN
1	2
2	1
...	...

(c) PUN1

BAM	EUH			
	3	4	5	3
1	3			
2	4			
3	5			
4	3			

(d) PUN2

BAM	EUH			
	1	2	6	
5	1			
6	2			
7	6			
8				

도면 13

(a) LBN-to-LUN

LBN	PBN
1	1
2	1
3	2
4	2
5	2
6	1
...	...

(b) LUN-to-PUN

LUN	PUN
1	2
2	3
...	...

(c) PUN1

BAM	EUH			
	3	4	5	3
1	3			
2	4			
3	5			
4	3			

(d) PUN3

BAM	EUH			
	3	4	5	
1	3			
2	4			
3	5			
4				

(d) PUN2

BAM	EUH			
	1	2	6	
5	1			
6	2			
7	6			
8				